UAEM | Universidad Autónoma del Estado de México

CENTRO UNIVERSITARIO UAEM
ZUMPANGO

Ingeniería en computación

Sistemas Operativos
Unidad de Competencia III

Ing. Diego Armando Ramírez Avelino

Contenido

- Esquemas de administración de memoria básica
- Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa
- Mecanismos de paginación
- Algoritmos de paginación

Objetivo

Comparar los esquemas y algoritmos de administración y asignación de memoria utilizando esquemas de paginación y segmentación.

Esquemas de administración

En la actualidad y con los avances tecnológicos el tamaño de la memoria se ha incrementado.

Recordemos algunos programas simples de clase y comparemos con algunos software específicos de la vida cotidiana.

Es importante conocer los diferentes esquemas de gestión de memoria, que pueden ser desde los más simples hasta los más sofisticados.

Algunos de los más simples aún se usan en algunos dispositivos móviles

Esquemas de administración

La gestión de memoria puede dividirse en dos clases:

- Los que mueven los procesos de la memoria principal al disco y del disco a la memoria principal durante su ejecución
- Y los que no lo hacen

Términos como el intercambio y la paginación son principalmente mecanismos artificiales motivados por la falta de memoria principal suficiente para contener todos los programas a la vez.

Por lo tanto si la memoria llegará a ser tan grande que siempre hubiera lo suficiente, los argumentos a favor de un tipo de esquema de gestión de memoria podrían volverse obsoletos.

Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa

Mono programación sin intercambio ni paginación

Este esquema de gestión consiste en ejecutar solo un programa a la vez, repartiendo la memoria entre ese programa y el sistema operativo.

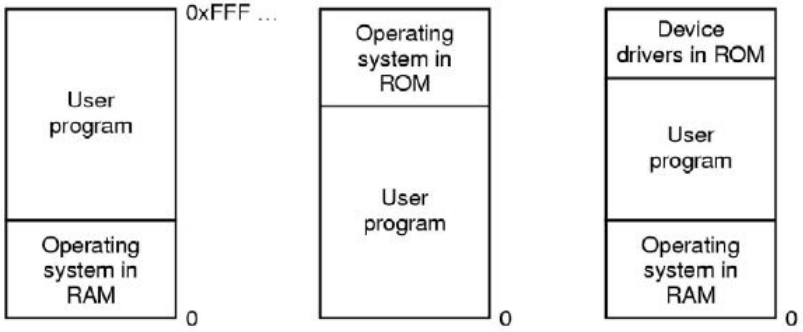
En la figura se muestran tres variaciones de este tema:

- En donde el sistema operativo podría estar en el fondo de la memoria RAM
- Podría estar en la ROM a lo alto de la memoria
- Los drivers de los dispositivos podrían estar en la parte más alta de la memoria ROM y el resto del sistema en la parte baja de la RAM

El primer modelo se utilizó antiguamente en mainframes y miniordenadores pero actualmente es muy raro su uso.

El segundo modelo se usa en algunos ordenadores palmtop y en sistemas empotrados

El tercer modelo se usó en los primeros ordenadores personales (ejecutando por ejemplo MS-DOS), donde la parte del sistema que está en ROM se denomina el **BIOS** (*Basic Input Output System*)



Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa

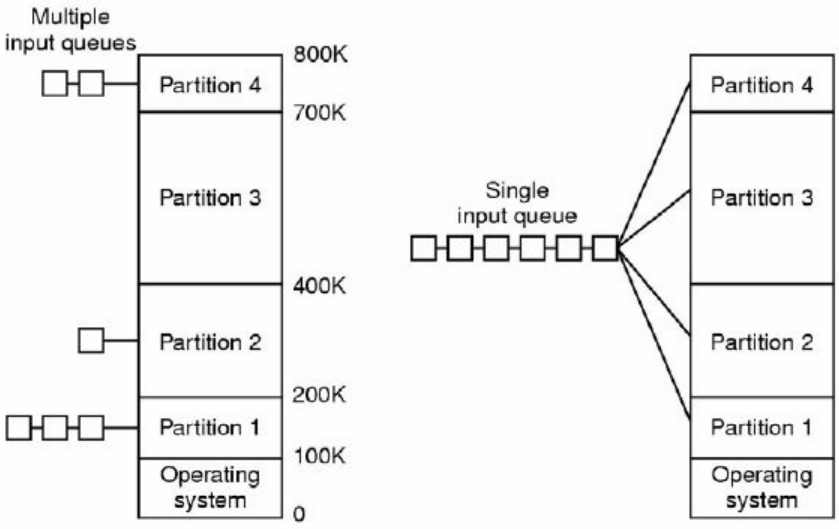
Multiprogramación con particiones fijas

La forma más fácil de conseguir la multiprogramación es simplemente dividir la memoria en *n* particiones (posiblemente de diferentes tamaños). Esta división puede, por ejemplo, realizarse manualmente cuando se pone en marcha el sistema

Este esquema de gestión actualmente se encuentra en desuso.

Porque la mayoría de los sistemas modernos permiten la ejecución de múltiples procesos al mismo tiempo. El tener múltiples procesos ejecutándose a la vez significa que cuando un proceso se bloquea esperando a que termine una operación de E/S, otro proceso puede seguir haciendo uso de la CPU

Así la Multiprogramación aumenta la utilización de los CPUs



Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa

Intercambio (*swapping*)

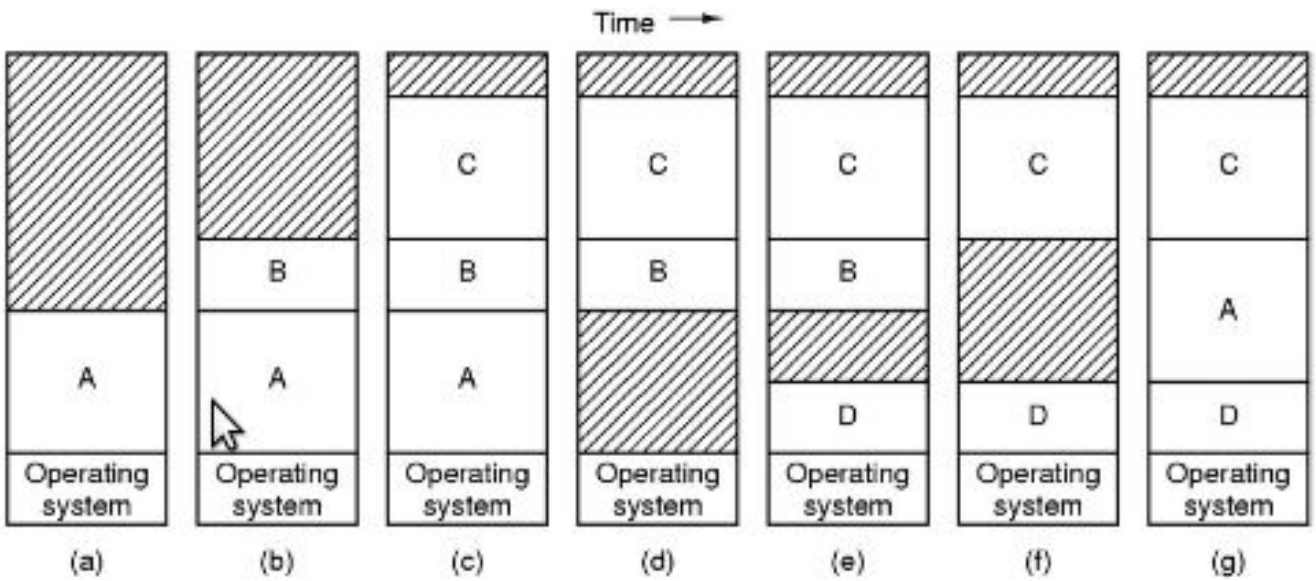
Con los sistemas de tiempo compartido o con los ordenadores personales orientados a gráficos A veces no hay suficiente memoria principal para contener a todos los procesos actualmente activos.

Por lo que los procesos de más deben mantenerse en el disco y cargarse en memoria para ejecutarse de forma dinámica

Una estrategia llamada intercambio consiste en cargar a la memoria un proceso entero, ejecutarlo durante un rato y volver a guardarlo en el disco.

Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa

Intercambio (*swapping*)



Métodos para llevar a cabo intercambio de información de memoria a disco y viceversa

Intercambio (*swapping*)

Es probable que la mayoría de los procesos crezcan durante su ejecución, y resulta una buena idea asignar un poco de memoria extra para que se intercambie.

Por ese motivo existen dos formas de Gestionar la Memoria de intercambio como:

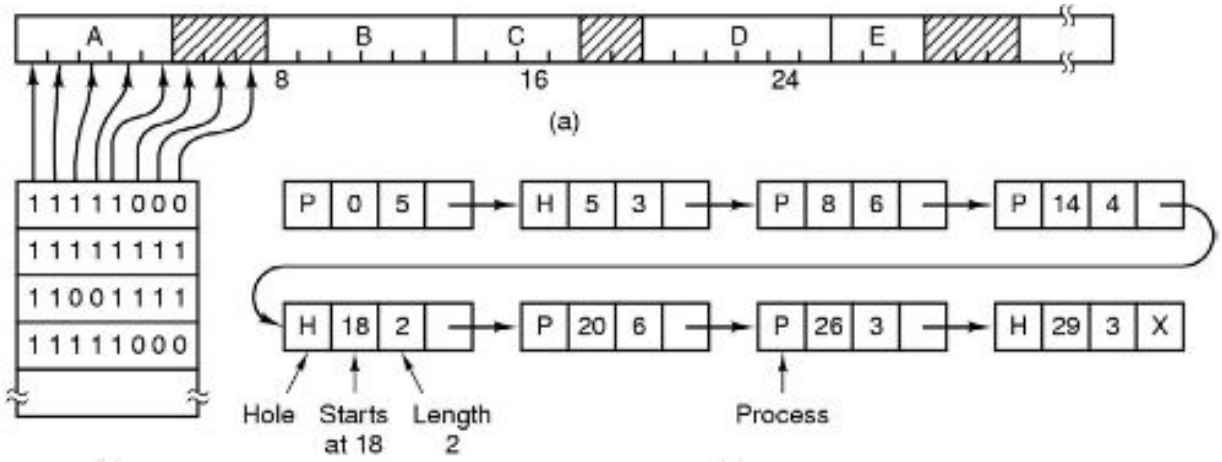
- usando mapas de bits
- usando listas enlazadas

Mecanismos de paginación

Gestión de memoria con mapas de bits

Con un mapa de bits, la memoria se divide en unidades de asignación, que pueden ser desde unas cuantas palabras hasta varios kilobytes.

A cada unidad de asignación le corresponde un bit del mapa de bits. El bit es 0 si la unidad de asignación está libre y 1 si está ocupada.



Mecanismos de paginación

Gestión de memoria con mapas de bits

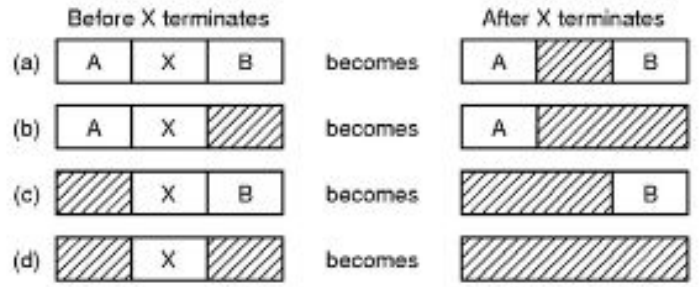
El tamaño de la unidad de asignación es una cuestión de diseño importante

Cuanto más pequeña sea la unidad, mayor será el mapa de bits.

Si se escoge una unidad de asignación grande, el mapa de bits será pequeño, pero podría desperdiciarse una unidad de memoria apreciable en la última unidad de asignación del proceso si el tamaño del proceso no es un múltiplo exacto de la unidad de asignación

Mecanismos de paginación

Gestión de memoria con listas enlazadas



Esta es otra forma de llevar control de la memoria donde principalmente se trata de mantener una lista enlazada de bloques de memoria asignados y libres, donde cada bloque es un proceso o un hueco entre dos procesos.

Normalmente un proceso que termina tiene dos procesos vecinos.

Dichos vecinos pueden ser procesos o hueco, lo que da lugar a las cuatro combinaciones.

- La actualización de la lista requiere sustituir una P por una H
- La unión de entradas en una sola
- La fusión de las entradas

Mecanismos de paginación

Gestión de memoria con listas enlazadas

Si los procesos y huecos se mantienen en una lista ordenada por dirección, pueden utilizarse varios algoritmos para asignar memoria a un proceso recién creado (o a un proceso existente que se intercambia a memoria).

Existen algoritmos para determinar por medio del gestor de memoria cuánta memoria se debe asignar al proceso.

- Algoritmo de primer ajuste (first fit)
- Algoritmo de segundo ajuste (second fit)
- Algoritmo de mejor ajuste (best fit)
- Algoritmo de peor ajuste (worst fit)

Los cuatro algoritmos anteriores pueden acelerarse manteniendo listas separadas para los procesos y los huecos.

El precio de esta aceleración es una complejidad adicional y ralentización cuando se libera la memoria, ya que los segmentos liberados deben eliminarse de la lista de procesos e insertarse en la lista de huecos.

Mecanismos de paginación

Gestión de memoria con listas enlazadas

Otro algoritmo de asignación es el de ajuste rápido (quick fit), que mantiene listas separadas para algunos de los tamaños solicitados más frecuentemente.

Mecanismos de paginación

Memoria Virtual

El método ideado por Fotheringham en 1961 se conoce ahora como memoria virtual.

La idea básica detrás de la memoria virtual es que el tamaño combinado del programa, sus datos y su pila pueden exceder la cantidad de memoria física disponible.

El sistema operativo mantiene en la memoria principal aquellas partes del programa que se están usando en cada momento, manteniendo el resto de las partes del programa en el disco.

Mecanismos de paginación

Paginación

La mayoría de los sistemas con memoria virtual usan una técnica denominada paginación.

Cuando el programa utiliza una instrucción como:

MOV REG, 1000

Estas direcciones generadas por el programa se denominan direcciones virtuales y constituyen el espacio de direcciones virtual.

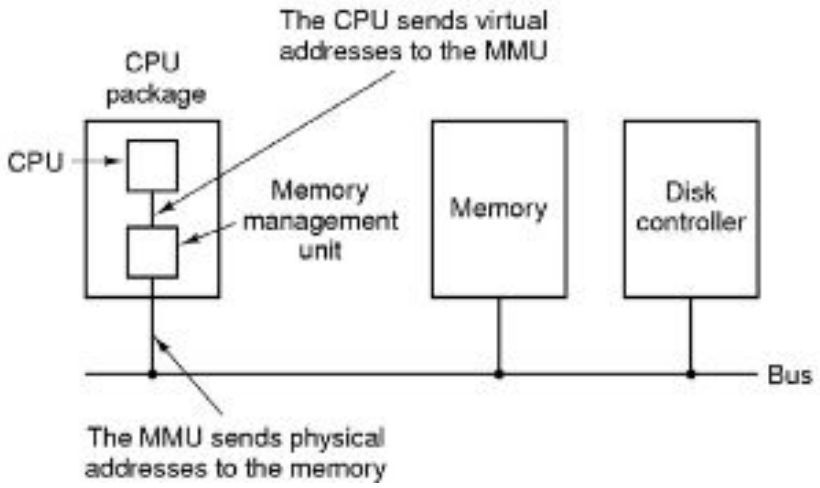
En ordenadores sin memoria virtual, la dirección virtual se coloca directamente sobre el bus de memoria y eso hace que la palabra de memoria física con esa dirección se lea o escriba.

Cuando se utiliza memoria virtual, las direcciones virtuales no se envían directamente al bus de memoria, sino que van a una unidad de Gestión de memoria (MMU), que establece una correspondencia entre las direcciones virtuales y las direcciones físicas de la memoria.

Mecanismos de paginación

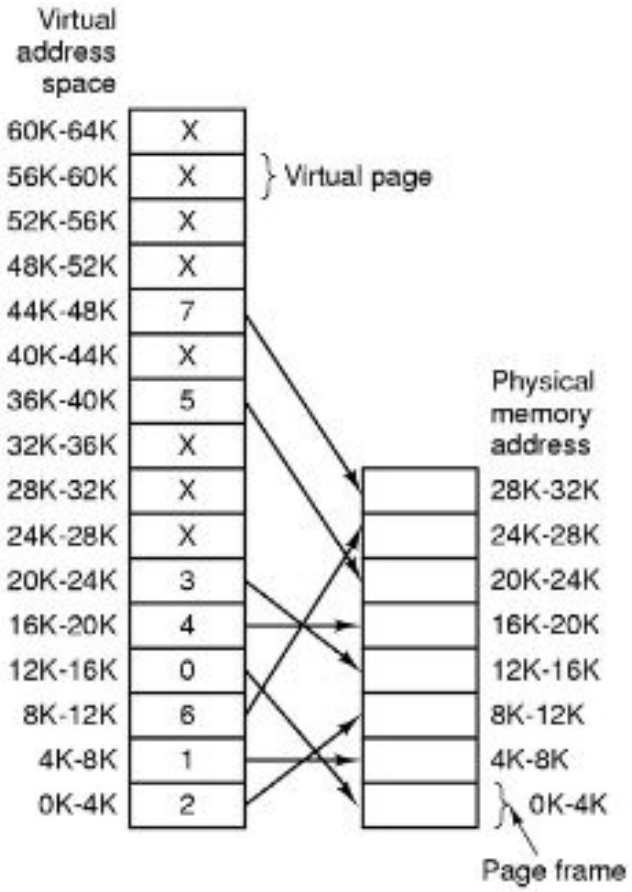
Paginación

En la figura se muestra la posición y función del MMU. Observe que es parte del chip de la CPU por que es lo más común en la actualidad.



Mecanismos de paginación

Paginación



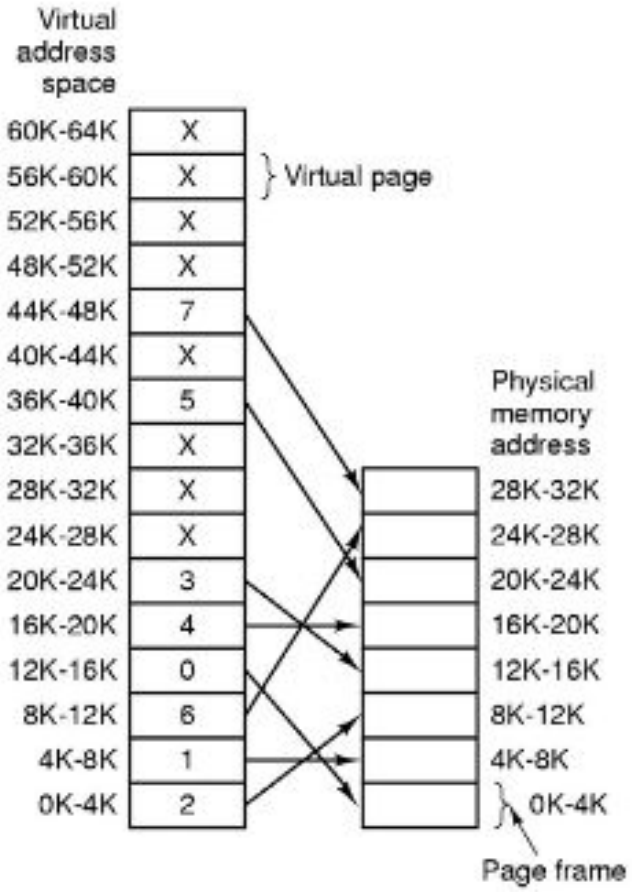
El espacio de direcciones virtual se divide en unidades llamadas páginas.

Las unidades correspondientes en la memoria física se denominan marcos de página.

Y la relación entre las direcciones virtuales y las direcciones de la memoria física viene dada por la tabla de páginas.

Mecanismos de paginación

Paginación



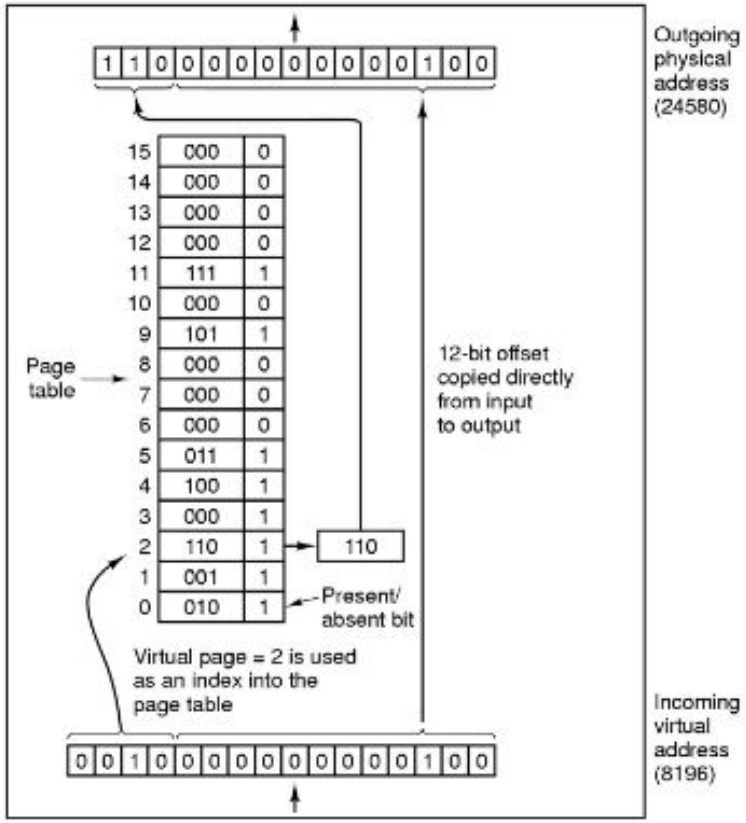
El número de página se utiliza como índice para consultar la tabla de páginas, obteniendo el número del marco de página correspondiente a esa página virtual.

Si el bit de presencia es 0, se generará una excepción que cederá el control al sistema operativo.

Si el bit es 1, el número de marco de página encontrado en la tabla de paginas se copia en los tres bits de mayor orden del registro de salida junto con el desplazamiento de 12 bits que se copia (sin ninguna modificación) de la dirección virtual recibida.

Mecanismos de paginación

Paginación



Juntos, esos dos campos forman una dirección física de 15 bits.

Finalmente, el registro de salida se vuelca al bus de memoria como la dirección de memoria física a la que efectivamente se va a acceder.

Mecanismos de paginación

Tablas de páginas

El propósito de la tabla de páginas es establecer una correspondencia aplicando las páginas virtuales sobre los marcos de página.

Matemáticamente, la tabla de páginas es una función, con el número de página virtual como argumento y el número de marco de página como resultado.

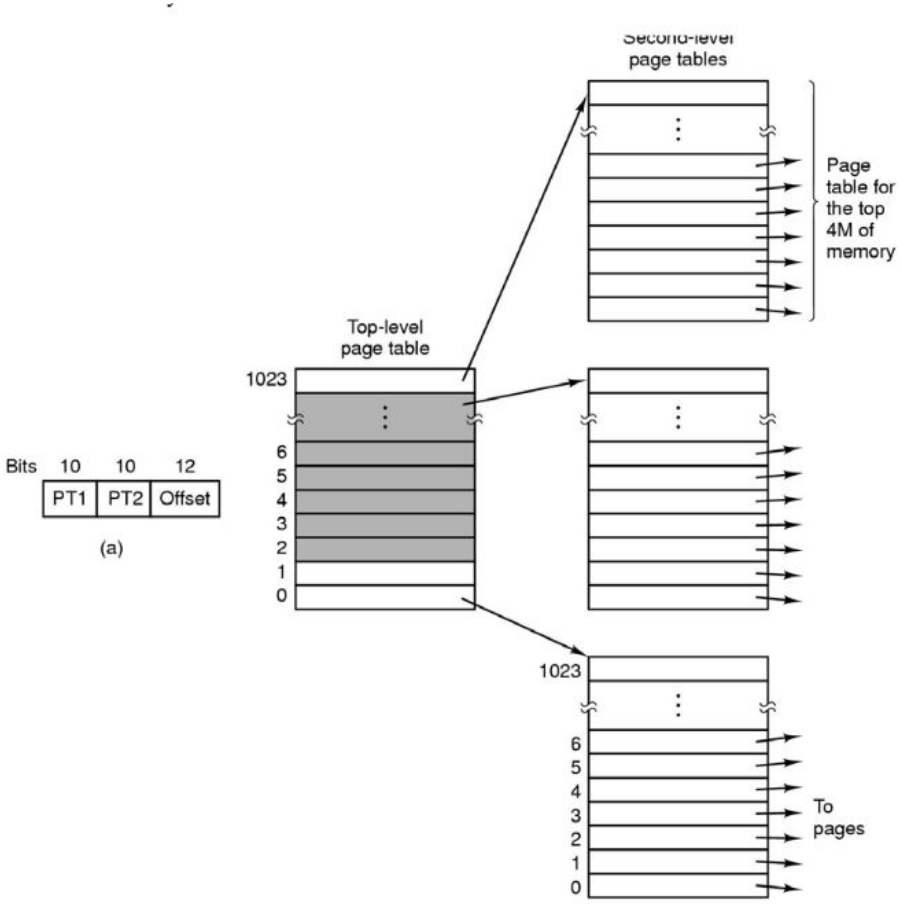
Utilizando el resultado de esta función, el campo de página virtual de una dirección virtual puede reemplazarse por un campo de marco de página, formando así una dirección de memoria física.

A pesar de lo sencillo de esta descripción, hay que resolver los siguientes problemas:

- La tabla de páginas puede ser extremadamente grande.
- La traducción de direcciones debe realizarse muy rápidamente

Mecanismos de paginación

Tablas de páginas multinivel



Mecanismos de paginación

Estructura de una Entrada de la Tabla de Páginas

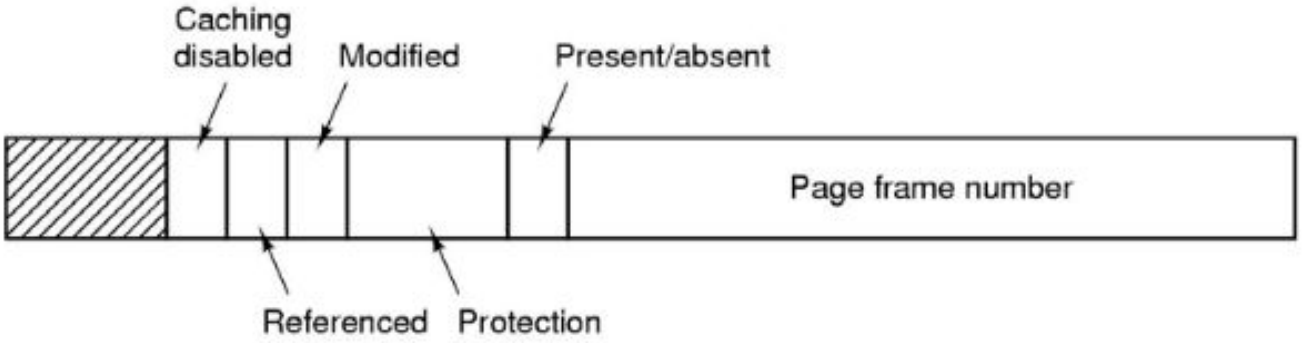
La organización exacta de una entrada depende mucho de la máquina, pero el tipo de información presente es casi el mismo en todos los ordenadores.

El campo más importante es el Número de marco de página. Después de todo, la función de la tabla de páginas es permitir encontrar ese valor.

El bit de Presencia. Si este bit es 1, la entrada es válida y puede usarse; si es 0, la página virtual a la que corresponde la entrada no está actualmente en memoria. Acceder a una entrada de la tabla de páginas con el bit de presencia a 0 provoca una falta de página.

Los bits de Protección indican qué tipos de acceso están permitidos. En su forma más simple, este campo contiene un bit, que vale 0 si se permite leer y escribir, o 1 si sólo se permite leer. Un esquema más sofisticado utiliza 3 bits, para habilitar/inhibir independientemente la lectura, la escritura y la ejecución de palabras de la página (análogo a los bits rwx).

Los bits de página Modificada y Referenciada siguen la pista del uso de la página. Cuando se escribe en una página, el hardware activa automáticamente el bit de Modificada.

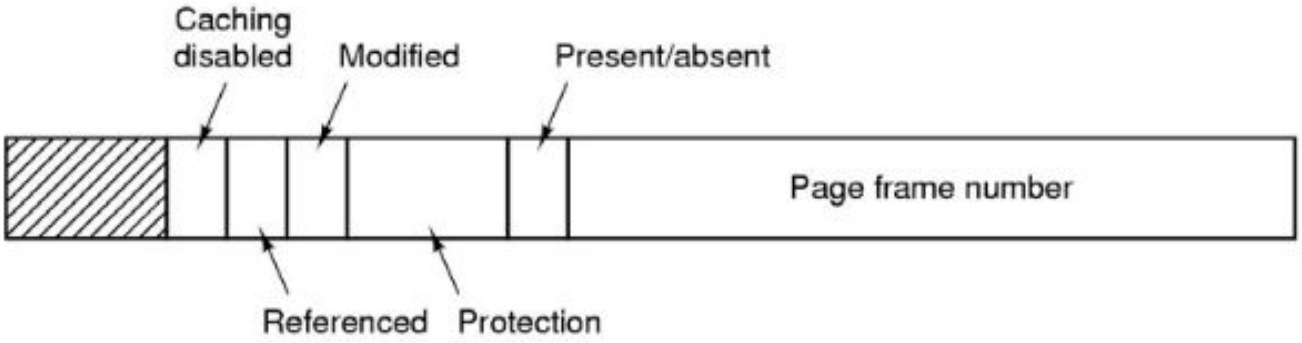


Mecanismos de paginación

Estructura de una Entrada de la Tabla de Páginas

El bit de Referencia se activa siempre que se referencia una página, ya sea para leer o para escribir. Su función es la de ayudar al sistema operativo a seleccionar la página que elegirá como víctima cuando se presente una falta de página.

Finalmente el último bit permite inhabilitar el uso de la caché para la página. Esta característica es importante en el caso de páginas que contienen direcciones correspondientes a registros de dispositivos, en vez de a posiciones de memoria. Si el sistema operativo está dando vueltas en un bucle de polling esperando a que algún dispositivo de E/S responda a un comando que se le acaba de enviar, es indispensable que el hardware siga extrayendo la palabra del dispositivo, y que no utilice una copia antigua almacenada en la caché. Con este bit, puede desactivarse el uso de la caché para esa página. Las máquinas que tienen un espacio de E/S separado y no utilizan E/S mapeada en memoria no necesitan ese bit.



Algoritmos de paginación

Existen 6 algoritmos de paginación para la administración de memoria:

- Algoritmo de sustitución de página óptimo
- De sustitución de páginas no usadas recientemente
- De sustitución de página de primera que entra primera que sale (FIFO),
- De sustitución de página de segunda oportunidad, de sustitución de página por reloj, y por último
- La de sustitución de página menos recientemente usada (LRU).

Algoritmos de paginación

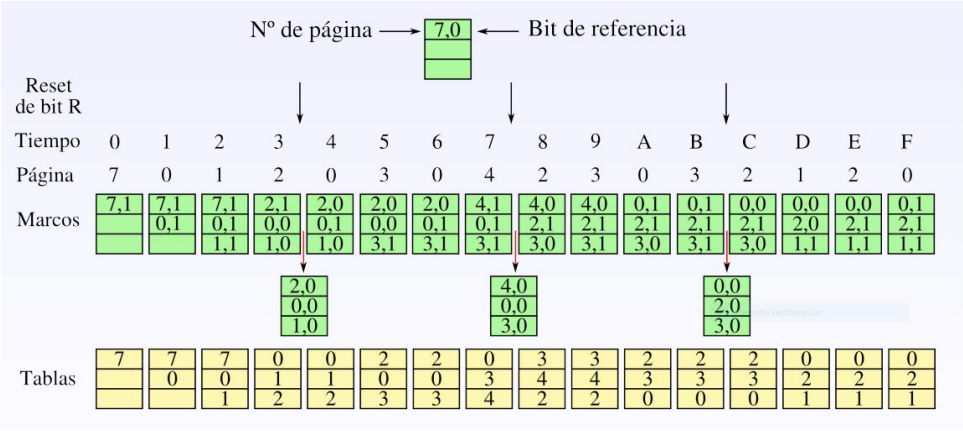
El algoritmo de sustitución de página óptimo tiene como característica principal eliminar la pagina que tenga el rótulo más alto, trata de aplazar los sucesos desagradables el mayor tiempo que se pueda y es fácil de describir pero imposible de implementar.

El algoritmo de sustitución de páginas no usadas recientemente se aplica cuando ocurre una falla de página, el sistema operativo examina todas las páginas y las divide en 4 categorías con base a sus valores. También supone que es mejor eliminar una pagina modificada a la que por lo menos no se ha hecho referencia en por lo menos un tic del reloj que una página limpia que no se está usando.

Algoritmos de paginación

El algoritmo de sustitución de página de primera que entra, primera que sale (FIFO); es de paginación con bajo consumo el sistema operativo mantiene una lista de todas las páginas que están en la memoria, si en la memoria existe la pagina que esta en la cabeza de la lista se le indica que es la más vieja y del final, es lo más reciente.

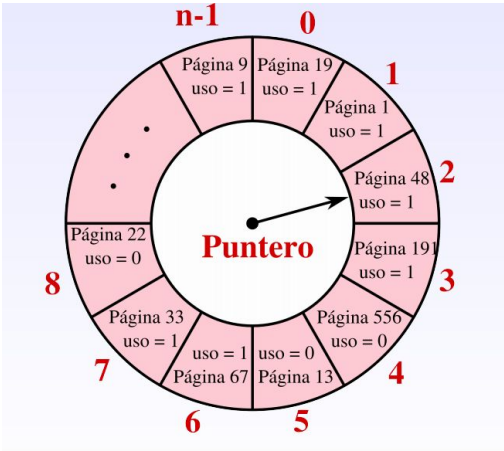
El algoritmo de sustitución de página de segunda oportunidad consiste en buscar una página vieja a la que no se haya hecho referencia en el intervalo de reloj. Si se ha hecho referencia a todas las páginas, este algoritmo pasa a ser FIFO puro.



Algoritmos de paginación

El algoritmo de sustitución de página por reloj, este mantiene todas las páginas en una lista circular con forma de reloj, si una manecilla apunta a la pagina mas vieja esta es sustituye el valor que se encontraba.

El algoritmo de sustitución de página menos recientemente usada (LRU); consiste en que si las páginas que se han usado mucho en las últimas instrucciones probablemente se usarán mucho en las siguientes. Se aplica cuando ocurre una falla de página este desaloja la página que haya estado más tiempo sin usarse. El sistema operativo examina todos los contadores de la tabla de páginas hasta encontrarse con el más bajo.



Bibliografía

- Carretero, P. (2001). Sistemas operativos: Una visión aplicada. Primera Edición.Madrid, Editorial McGraw-Hill.
- Deitel, M. (1993). Introducción a los sistemas operativos. Segunda Edición México, Editorial Addison Wesley Longman de México,.
- Dhamdhere, D. (2008). Sistemas operativos. Segunda Edición. México, Editorial McGraw-Hill.
- Flynn, M. (2001). Sistemas operativos. Tercera Edición. México, Editorial International Thomson.
- Galli, D. (2000). Distributed operating systems. New Jersey, Editorial Prentice-Hall.
- Mclver McHoes, A. (2011). Sistemas operativos. Sexta Edición. México, Editorial CENGAGE Learning.
- Pérez, F. (2003). Problemas de sistemas operativos: de la base al diseño. Segunda Edición. Madrid. Editorial McGraw-Hill.
- Ramez, E. (2010). Operating Systems: A Spiral Approach. Primera Edición. México, Editorial McGraw-Hill.
- Silberschatz, A. (2006). Fundamentos de sistemas operativos. Séptima Edición. Madrid, Editorial McGraw-Hill.
- Silberschatz, A. (2008). Sistemas operativos. Sexta Edición. México, Editorial Limusa.
- Stallings, W. (1995). Sistemas operativos. Sexta Edición. México, Editorial Limusa.


HUMANISMO QUE TRANSFORMA
2013-2017